BS ISO/IEC 30108-2:2023

BSI Standards Publication

Biometrics — Identity attributes verification services

Part 2: RESTful specification

Documents
Document Sharing Hub

Full document details: BS ISO/IEC 30108-2:2023

National foreword

This British Standard is the UK implementation of ISO/IEC 30108-2:2023.

The UK participation in its preparation was entrusted to Technical Committee IST/44/-/2, Biometric technical interfaces.

A list of organizations represented on this committee can be obtained on request to its committee manager.

Contractual and legal considerations

This publication has been prepared in good faith, however no representation, warranty, assurance or undertaking (express or implied) is or will be made, and no responsibility or liability is or will be accepted by BSI in relation to the adequacy, accuracy, completeness or reasonableness of this publication. All and any such responsibility and liability is expressly disclaimed to the full extent permitted by the law.

This publication is provided as is, and is to be used at the recipient's own risk.

The recipient is advised to consider seeking professional guidance with respect to its use of this publication.

This publication is not intended to constitute a contract. Users are responsible for its correct application.

© The British Standards Institution 2023
Published by BSI Standards Limited 2023

ISBN 978 0 539 05962 5

ICS 35.240.15

Compliance with a British Standard cannot confer immunity from legal obligations.

This British Standard was published under the authority of the Standards Policy and Strategy Committee on 30 November 2023.

Amendments/corrigenda issued since publication

Date	Text affected

Documents
Document Sharing Hub

Full document details: BS ISO/IEC 30108-2:2023

INTERNATIONAL STANDARD

ISO/IEC 30108-2

First edition
2023-10

Biometrics — Identity attributes verification services —

Part 2: RESTful specification

Biométrique — Services de vérification des attributs d'identité — Partie 2: Spécification RESTful

Reference number
ISO/IEC 30108-2:2023(E)

Documents
Document Sharing Hub
© ISO/IEC 2023

Full document details: BS ISO/IEC 30108-2:2023

COPYRIGHT PROTECTED DOCUMENT

© ISO/IEC 2023

All rights reserved. Unless otherwise specified, or required in the context of its implementation, no part of this publication may be reproduced or utilized otherwise in any form or by any means, electronic or mechanical, including photocopying, or posting on the internet or an intranet, without prior written permission. Permission can be requested from either ISO at the address below or ISO's member body in the country of the requester.

ISO copyright office
CP 401 • Ch. de Blandonnet 8
CH-1214 Vernier, Geneva
Phone: +41 22 749 01 11
Email: copyright@iso.org
Website: www.iso.org

Published in Switzerland

© ISO/IEC 2023 – All rights reserved

Documents
Document Sharing Hub

Full document details: BS ISO/IEC 30108-2:2023

Contents

Page

Foreword ... **v**
Introduction ... **vi**
1 Scope ... **1**
2 Normative references ... **1**
3 Terms and definitions ... **1**
4 Symbols and abbreviated terms ... **2**
5 Conformance ... **2**
6 System context ... **2**
7 Identity assurance services ... **3**
7.1 General ... 3
7.2 Primitive services ... 3
7.2.1 Introduction ... 3
7.2.2 Add Subject To Gallery ... 3
7.2.3 Check Quality ... 5
7.2.4 Classify Biometric Data ... 7
7.2.5 Create Encounter ... 9
7.2.6 Create Subject ... 11
7.2.7 Delete Biographic Data ... 12
7.2.8 Delete Biometric Data ... 14
7.2.9 Delete Document Data ... 16
7.2.10 Delete Encounter ... 17
7.2.11 Delete Subject ... 19
7.2.12 Delete Subject From Gallery ... 20
7.2.13 Get Identify Subject Results ... 22
7.2.14 Identify Subject ... 23
7.2.15 List Biographic Data ... 26
7.2.16 List Biometric Data ... 28
7.2.17 List Document Data ... 30
7.2.18 Perform Fusion ... 33
7.2.19 Query Capabilities ... 34
7.2.20 Retrieve Biographic Data ... 35
7.2.21 Retrieve Biometric Data ... 37
7.2.22 Retrieve Document Data ... 39
7.2.23 Set Biographic Data ... 41
7.2.24 Set Biometric Data ... 43
7.2.25 Set Document Data ... 46
7.2.26 Transform Biometric Data ... 48
7.2.27 Update Biographic Data ... 50
7.2.28 Update Biometric Data ... 52
7.2.29 Update Document Data ... 54
7.2.30 Verify Subject ... 55
7.3 Aggregated Services ... 58
7.3.1 Delete ... 58
7.3.2 Enrol ... 60
7.3.3 Get Deletion Results ... 63
7.3.4 Get Enrol Results ... 64
7.3.5 Get Identify Results ... 66
7.3.6 Get Update Results ... 68
7.3.7 Get Verify Results ... 69
7.3.8 Identify ... 71
7.3.9 Retrieve Data ... 74
7.3.10 Update ... 76

© ISO/IEC 2023 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO/IEC 30108-2:2023

7.3.11 Verify ... 78

8 Data elements and data types ... 82
8.1 Introduction ... 82
8.2 Biographic data ... 82
8.2.1 General ... 82
8.2.2 Biographic Data Item Type ... 82
8.2.3 Biographic Data List Type ... 83
8.2.4 Biographic Data Set Type ... 83
8.2.5 Biographic Data Type ... 84
8.3 Biometric Data ... 84
8.3.1 General ... 84
8.3.2 Biometric Data Element Type ... 84
8.3.3 Biometric Data List Type ... 85
8.3.4 Biometric Type ... 85
8.3.5 CBEFF BIR Type ... 86
8.3.6 CBEFF BIR List Type ... 87
8.4 Candidate Lists ... 87
8.4.1 General ... 87
8.4.2 Candidate List Type ... 87
8.4.3 Candidate Type ... 87
8.5 Document Data ... 88
8.5.1 General ... 88
8.5.2 Document Data List Type ... 88
8.5.3 Document Data Type ... 89
8.6 Capabilities ... 90
8.6.1 Capability List Type ... 90
8.6.2 Capability Type ... 90
8.7 Fusion Information ... 101
8.7.1 Introduction ... 101
8.7.2 Fusion Identity List Type ... 101
8.7.3 Fusion Information List Type ... 101
8.7.4 Fusion Information Type ... 101
8.8 Other Data Types ... 102
8.8.1 Encounter Category Type ... 102
8.8.2 Encounter List Type ... 103
8.8.3 Information Type ... 103
8.8.4 List Filter Type ... 103
8.8.5 Option Type ... 104
8.8.6 Processing Options Type ... 104
8.8.7 Token Type ... 105

9 Error handling and notification ... 105
9.1 Introduction ... 105
9.2 Generic HTTP responses ... 105
9.3 Error condition codes ... 106
9.4 YAML specification ... 109

10 Security ... 118

Annex A (normative) **OpenAPI™ specification in YAML ... 119**

Annex B (normative) **CBEFF Patron Format in YAML ... 121**

Bibliography ... 126

© ISO/IEC 2023 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO/IEC 30108-2:2023

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work.

The procedures used to develop this document and those intended for its further maintenance are described in the ISO/IEC Directives, Part 1. In particular, the different approval criteria needed for the different types of document should be noted. This document was drafted in accordance with the editorial rules of the ISO/IEC Directives, Part 2 (see www.iso.org/directives or www.iec.ch/members_experts/refdocs).

ISO and IEC draw attention to the possibility that the implementation of this document may involve the use of (a) patent(s). ISO and IEC take no position concerning the evidence, validity or applicability of any claimed patent rights in respect thereof. As of the date of publication of this document, ISO and IEC had not received notice of (a) patent(s) which may be required to implement this document. However, implementers are cautioned that this may not represent the latest information, which may be obtained from the patent database available at www.iso.org/patents and https://patents.iec.ch. ISO and IEC shall not be held responsible for identifying any or all such patent rights.

Any trade name used in this document is information given for the convenience of users and does not constitute an endorsement.

For an explanation of the voluntary nature of standards, the meaning of ISO specific terms and expressions related to conformity assessment, as well as information about ISO's adherence to the World Trade Organization (WTO) principles in the Technical Barriers to Trade (TBT) see www.iso.org/iso/foreword.html. In the IEC, see www.iec.ch/understanding-standards.

This document was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 37, *Biometrics*.

A list of all parts in the ISO/IEC 30108 series can be found on the ISO and IEC websites.

Any feedback or questions on this document should be directed to the user's national standards body. A complete listing of these bodies can be found at www.iso.org/members.html and www.iec.ch/national-committees.

© ISO/IEC 2023 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO/IEC 30108-2:2023

Introduction

This document defines the architecture, operations, data elements and basic requirements for Identity Attributes Verification Services (IAVSs), thereby providing a framework for the implementation of generic identity services within a service-oriented environment. An identity in the context of IAVS comprises a subject, biographic data and biometric data. Other parts of the ISO/IEC 30108 series are intended to define specific IAVS implementations (or bindings) within specific environments, for example, Simple Object Access Protocol (SOAP) web services.

IAVS services are generic in nature, being modality-neutral and not targeted at any particular business application. These services include those related to the management, transformation and biometric comparison identity data. Services are invoked by an IAVS requester and implemented by an IAVS service provider (responder). IAVS does not prescribe the architecture or business logic of either the requester or service provider.

In IAVS two categories of identity services are defined: primitive and aggregate. Primitive services are more atomic and well-defined, whereas the aggregate services tend to be higher level and enable more flexibility on the part of the IAVS service provider.

In IAVS two identity models are also defined: person-centric and encounter-based. Person-centric systems maintain a single up-to-date record (set of data) for a given person, whereas an encounter-based system retains data related to each interaction the person has with the system.

This document represents a version of IAVS defined in ISO/IEC 30108-1, but using a representational state transfer (RESTful) approach.

© ISO/IEC 2023 – All rights reserved
Documents
Document Sharing Hub
Full document details: BS ISO/IEC 30108-2:2023

Biometrics — Identity attributes verification services —

Part 2: RESTful specification

1 Scope

The ISO/IEC 30108 series defines biometric services used for identity assurance that are invoked over a services-based framework. It provides a generic set of biometric and identity-related functions and associated data definitions to allow remote access to biometric services.

Although focused on biometrics, the ISO/IEC 30108 series includes support for other related identity assurance mechanisms such as biographic and document capabilities. Identity attributes verification services (IAVSs) are intended to be compatible with and used in conjunction with other biometric standards as described in ISO/IEC 30108-1.

This document implements the specification provided in ISO/IEC 30108-1 using representational state transfer (REST).

Specification of biometric functionality is limited to remote (backend) services. Services between a client-side application and biometric capture devices are not within the scope of this document.

Integration of biometric services as part of an authentication service or protocol is not within the scope of this document.

2 Normative references

The following documents are referred to in the text in such a way that some or all of their content constitutes requirements of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

ISO/IEC 2382-37, *Information technology — Vocabulary — Part 37: Biometrics*

ISO/IEC 19785-1, *Information technology — Common Biometric Exchange Formats Framework — Part 1: Data element specification*

ISO/IEC 30108-1:2015[1)], *Information technology — Biometric Identity Assurance Services — Part 1: BIAS services*

3 Terms and definitions

For the purposes of this document, the terms and definitions given in ISO/IEC 30108-1 and ISO/IEC 2382-37 apply.

ISO and IEC maintain terminology databases for use in standardization at the following addresses:

— ISO Online browsing platform: available at https://www.iso.org/obp

— IEC Electropedia: available at https://www.electropedia.org/

1) Under revision.

© ISO/IEC 2023 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO/IEC 30108-2:2023

4 Symbols and abbreviated terms

For the purposes of this document, the symbols and abbreviated terms given in ISO/IEC 30108-1 and the following apply.

BIR	biometric information record
BRA	ISO Biometric Registration Authority
CBEFF	Common Biometric Exchange Formats Framework (defined in ISO/IEC 19785 series)
EFTS	electronic fingerprint transmission specification
ID	identity / identification / identifier
IAVS	identity attributes verification services
JSON	JavaScript Object Notation
REST	representational state transfer
SOAP	Simple Object Access Protocol
UUID	universally unique identifier
YAML	Yet Another Markup Language[3]

5 Conformance

ISO/IEC 30108-1:2015, Annex A specifies the conformance requirements for systems/components claiming conformance to this document.

6 System context

This clause provides an overview of representational state transfer (REST), in the scope of IAVS.

The specification included in this document has been written in OpenAPI™,[2)] using YAML™ (Yet Another Markup Language)[3)] as the specification language. OpenAPI is an extended way to specify RESTful services, allowing a variety of tools to develop client and server applications, as well as using data in any of the data formats typically used in RESTful services, e.g. JSON (JavaScript Object Notation) or XML (eXtensible Markup Language).

A goal for this document is to be as open as possible. To that end, this specification is also available in an electronic OpenAPI file, available at https://standards.iso.org/iso-iec/30108/-2/ed-1.

The specification is written according to the following references:

— ECMA Standard ECMA-404 (2017 2nd Edition) The JSON Data Interchange Format ISO\IEC 21778

— Hyper Text Transfer Protocol (HTTP/1.1) RFC 7230 [https://httpwg.org/specs/rfc7230.html]

— Hypertext Transfer Protocol (HTTP/1.1): Semantics and Content RFC 7231

— JSON Schema: A Media Type for Describing JSON Documents [http://json-schema.org/draft/2019-09/json-schema-core.html]

2) This trademark is provided for reasons of public interest or public safety. This information is given for the convenience of users of this document and does not constitute an endorsement by ISO or IEC.

3) This trademark is provided for reasons of public interest or public safety. This information is given for the convenience of users of this document and does not constitute an endorsement by ISO or IEC.

© ISO/IEC 2023 – All rights reserved
Documents
Document Sharing Hub
Full document details: BS ISO/IEC 30108-2:2023

— YAML Ain't Markup Language (YAML™) Version 1.2: 3rd Edition, Patched at 2009-10-01 [https://yaml.org/spec/1.2/spec.html]

All subclauses within Clause 7 entitled "REST method" (numbered 7.x.x.2) indicate the HTTP method used, among the various possibilities: GET, POST, DELETE, PUT and PATCH.

Within this document, when a YAML specification is given (e.g. all subclauses numbered 7.x.x.5), it shall be placed within an OpenAPI file, in the section indicated, taking into account the template provided in Annex A.

7 Identity assurance services

7.1 General

This clause defines two categories of IAVS services: primitive and aggregate. Primitive services are lower-level operations that are used to request a specific capability. Aggregate services operate at a higher-level, performing a sequence of primitive operations in a single request. (An example of such a sequence would be a negative search where a 1:N identification which results in no matches being found is immediately followed by the addition of the biometric sample into that search population.) Implementers are not restricted to one or the other but they may provide (and requesters may use) a mixture of both primitive and aggregate types.

Aggregate services do not have to utilize primitive services.

Each service is identified by an *<interface>* tag and must include a *name* attribute. Service parameters are identified by a *<parameter>* tag and must include a *name*, *type* and *direction* attribute. The *direction* attribute specifies whether the parameter is an input parameter (*in*), an output parameter (*out*) or an input/output parameter (*inout*). Parameters may also include a *use* attribute to indicate if the parameter is required, optional or conditional. If the parameter is conditional, the service description must identify the conditions.

7.2 Primitive services

7.2.1 Introduction

IAVS specifies the following set of primitive services.

7.2.2 Add Subject To Gallery

7.2.2.1 Description

The *Add Subject To Gallery* service shall register a subject to a given gallery or population group. As an optional parameter, the value of the claim to identity by which the subject is known to the gallery may be specified. This claim to identity shall be unique across the gallery. If no claim to identity is specified, the subject ID (assigned with the *Create Subject* service) shall be used as the claim to identity. Additionally, in the encounter-centric model, the encounter ID associated with the subject's biometric sample that will be added to the gallery shall be specified.

NOTE In the IAVS model, the creation and management of galleries are the responsibility of the service provider implementation. Services are not exposed to the requester for this purpose.

7.2.2.2 REST method

POST

© ISO/IEC 2023 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO/IEC 30108-2:2023

7.2.2.3 Parameters

— *Gallery ID* — the identifier of the gallery or population group to which the subject will be added.

— *Subject ID* — the identifier of the subject.

— *Identity Claim (optional)* — the identifier by which the subject is known to the gallery.

— *Encounter ID (conditional)* — the identifier of the encounter, required for encounter-centric models.

7.2.2.4 Responses

In addition to the generic HTTP responses defined in 9.2, the execution of this service shall provide one of the responses listed in Table 1 (see 9.3 for the definitions of each error condition code).

Table 1 — Particular responses for *Add Subject To Gallery* service

HTTP status code	HTTP status code meaning	Error condition code	Description
200	OK	SUCCESS	
400	Bad Request	INVALID_IDENTITY_CLAIM	
		INVALID_SUBJECT_ID	
		INVALID_ENCOUNTER_ID	
404	Not Found	UNKNOWN_GALLERY	
		UNKNOWN_SUBJECT	
		UNKNOWN_IDENTITY_CLAIM	
		UNKNOWN_ENCOUNTER	
500	Internal Server Error	CANNOT_STORE_DATA	
		INTERNAL_DATABASE_ERROR	

7.2.2.5 YAML specification

To be placed in section #/paths:

```
/addSubjectToGallery:
    post:
      summary: Add Subject To Gallery
      parameters:
        - name: galleryID
          in: query
          required: true
          schema:
            type: string
        - name: subjectID
          in: query
          required: true
          schema:
            type: string
        - name: identityClaim
          in: query
          required: false
          schema:
            type: string
        - name: encounterID
          in: query
          required: false #conditional
          schema:
            type: string
      responses:
        200:
          $ref: '#/components/responses/success'
```


© ISO/IEC 2023 – All rights reserved
Documents
Document Sharing Hub
Full document details: BS ISO/IEC 30108-2:2023

```
        400:
          description: Bad Request
          content:
            application/json:
              schema:
                oneOf:
                  - $ref: '#/components/schemas/invalidIdentityClaim'
                  - $ref: '#/components/schemas/invalidSubjectId'
                  - $ref: '#/components/schemas/invalidEncounterId'
        404:
          description: Not Found
          content:
            application/json:
              schema:
                oneOf:
                  - $ref: '#/components/schemas/unknownGallery'
                  - $ref: '#/components/schemas/unknownSubject'
                  - $ref: '#/components/schemas/unknownIdentityClaim'
                  - $ref: '#/components/schemas/unknownEncounter'
        500:
          description: Internal Server Error
          content:
            application/json:
              schema:
                oneOf:
                  - $ref: '#/components/schemas/cannotStoreData'
                  - $ref: '#/components/schemas/internalDatabaseError'
                  - $ref: '#/components/schemas/unknownError'
        503:
          $ref: '#/components/responses/x503'
```

7.2.3 Check Quality

7.2.3.1 Description

The *Check Quality* service shall return a quality score for a given biometric sample or a specified subject. Either a biometric sample or a subject identifier (ID) shall be provided. The biometric input is provided in a Common Biometric Exchange Formats Framework (CBEFF, as defined in ISO/IEC 19785 series) basic structure or CBEFF record, which in this document is called a CBEFF-BIR (Biometric Information Record). The algorithm vendor and algorithm vendor product ID may be optionally provided in order to request a particular algorithm's use in calculating the biometric quality. If an algorithm vendor is provided, then the algorithm vendor product ID is required. If no algorithm vendor is provided, the implementing system shall provide the algorithm vendor and algorithm vendor product ID that were used to calculate the biometric quality as output parameters.

NOTE Algorithm vendors are registered with the ISO Biometric Registration Authority (BRA) and assigned unique identifiers as defined in ISO/IEC 19785-2. Algorithm Product IDs are assigned by the registered algorithm vendor.

7.2.3.2 REST method

GET

7.2.3.3 Parameters

— *BIR (conditional)* — data structure containing a single biometric sample for which a quality score is to be determined; required if no Subject ID is provided.

— *Subject ID (conditional)* — the identifier of the subject; required if no BIR is provided.

— *Algorithm Vendor (optional)* — the identifier of the vendor of the quality algorithm used to determine the quality.

© ISO/IEC 2023 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO/IEC 30108-2:2023

— *Algorithm Vendor Product ID (conditional)* — the vendor-assigned ID for the algorithm used to determine the quality; required as input if algorithm vendor is provided.

7.2.3.4 Responses

In addition to the generic HTTP responses defined in 9.2, the execution of this service shall provide one of the responses listed in Table 2 (see 9.3 for the definitions of each error condition code).

Table 2 — Particular responses for *Check Quality* service

HTTP status code	HTTP status code meaning	Error condition code	Description
200	OK	SUCCESS	Adding also — *Quality Score* — *Algorithm Version* as seen below table.
400	Bad Request	INVALID_BIR	
		INVALID_SUBJECT_ID	
		INVALID_INPUT	
		BIOMETRIC_TYPE_NOT_SUPPORTED	
		UNKNOWN_FORMAT	
403	Forbidden	BIR_DECRYPTION_FAILURE	
		BIR_QUALITY_ERROR	
		BIR_SIGNATURE_FAILURE	
404	Not Found	UNKNOWN_SUBJECT	
500	Internal Server Error	CANNOT_CHECK_QUALITY	

A successful execution of this service will provide:

— *Quality Score* — the quality of the biometric, as defined by the Quality type in one of the JSON-coded patron formats ISO/IEC 19785-3.

— *Algorithm Version* — the version of the algorithm used to determine the quality

7.2.3.5 YAML specification

To be placed in section #/paths:

```
/checkQuality:
  get:
    summary: Check Quality
    parameters:
      - name: bir
        in: query
        required: false #conditional
        schema:
          $ref: '#/components/schemas/CBEFF_BIR_Type'
      - name: subjectID
        in: query
        required: false #conditional
        schema:
          type: string
      - name: algorithmVendor
        in: query #inout
        required: false
        schema:
          type: string
```


© ISO/IEC 2023 – All rights reserved
Documents
Document Sharing Hub
Full document details: BS ISO/IEC 30108-2:2023

```
      - name: algorithmVendorProductID
        in: query #inout
        required: false #conditional
        schema:
          type: string
    responses:
      200:
        description: Success. Returns quality score and algorithm version
        content:
          application/json:
            schema:
              type: object
              properties:
                qualityScore:
                  type: string #RAUL: CBEFF-3 JSON
                  description: >
                    The quality of the biometric, as defined by the Quality
                    type in one of the JSON-coded patron formats ISO/IEC
                    19785-3
                algorithmVersion:
                  type: string
                  description: >
                    The version of the algorithm used to determine the
                    quality
      400:
        description: Bad Request
        content:
          application/json:
            schema:
              oneOf:
                - $ref: '#/components/schemas/invalidBir'
                - $ref: '#/components/schemas/invalidSubjectId'
                - $ref: '#/components/schemas/invalidInput'
                - $ref: '#/components/schemas/biometricTypeNotSupported'
                - $ref: '#/components/schemas/unknownFormat'
      403:
        description: Forbidden
        content:
          application/json:
            schema:
              oneOf:
                - $ref: '#/components/schemas/birDecryptionFailure'
                - $ref: '#/components/schemas/birQualityError'
                - $ref: '#/components/schemas/birSignatureFailure'
      404:
        description: Unknown Subject
        content:
          application/json:
            schema:
              $ref: '#/components/schemas/unknownSubject'
      500:
        description: Internal Server Error
        content:
          application/json:
            schema:
              oneOf:
                - $ref: '#/components/schemas/cannotCheckQuality'
                - $ref: '#/components/schemas/unknownError'
      503:
        $ref: '#/components/responses/x503'
```

7.2.4 Classify Biometric Data

7.2.4.1 Description

The *Classify Biometric Data* service shall attempt to classify a biometric sample. For example, a fingerprint biometric sample may be classified as a whorl, loop or arch (or other classification classes and sub-classes). The types of classification algorithms and classes are not specified here. Instead, they

© ISO/IEC 2023 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO/IEC 30108-2:2023

are left for the implementing system to define. If no classification algorithm is input, then the IAVS service provider will make the selection.

7.2.4.2 REST method

GET

7.2.4.3 Parameters

— *BIR* — data structure containing a single biometric sample for which the classification is to be determined.

— *Classification Algorithm Type (optional/output)* — identifies the type of classification algorithm to be used (input) and that was used (output) to perform the classification (e.g. for fingerprints, Henry classification).

7.2.4.4 Responses

In addition to the generic HTTP responses defined in 9.2, the execution of this service shall provide one of the responses listed in Table 3 (see 9.3 for the definitions of each error condition code).

Table 3 — Particular responses for *Classify Biometric Data* service

HTTP status code	HTTP status code meaning	Error condition code	Description
200	OK	SUCCESS	Adding also *Classification*, as seen below table.
400	Bad Request	INVALID_BIR	
		INVALID_INPUT	
		BIOMETRIC_TYPE_NOT_SUPPORTED	
		UNKNOWN_FORMAT	
403	Forbidden	BIR_DECRYPTION_FAILURE	
		BIR_QUALITY_ERROR	
		BIR_SIGNATURE_FAILURE	
500	Internal Server Error	CANNOT_PROCESS_DATA	

A successful execution of this service will provide:

— *Classification* — the result of the classification.

7.2.4.5 YAML specification

To be placed in section #/paths:

```
/classifyBiometricData:
  get:
    summary: Classify Biometric Data
    parameters:
      - name: bir
        in: query
        required: true
        schema:
          $ref: '#/components/schemas/CBEFF_BIR_Type'
      - name: classificationAlgorithmType
        in: query
        required: false
        schema:
```


© ISO/IEC 2023 – All rights reserved
Documents
Document Sharing Hub
Full document details: BS ISO/IEC 30108-2:2023

```
        type: string
    responses:
      200:
        description: Success. Returns classification
        content:
          application/json:
            schema:
              type: object
              properties:
                classification:
                  type: string
                  description: The result of the classification
      400:
        description: Bad Request
        content:
          application/json:
            schema:
              oneOf:
                - $ref: '#/components/schemas/invalidBir'
                - $ref: '#/components/schemas/invalidInput'
                - $ref: '#/components/schemas/biometricTypeNotSupported'
                - $ref: '#/components/schemas/unknownFormat'
      403:
        description: Forbidden
        content:
          application/json:
            schema:
              oneOf:
                - $ref: '#/components/schemas/birDecryptionFailure'
                - $ref: '#/components/schemas/birQualityError'
                - $ref: '#/components/schemas/birSignatureFailure'
      500:
        description: Internal Server Error
        content:
          application/json:
            schema:
              oneOf:
                - $ref: '#/components/schemas/cannotProcessData'
                - $ref: '#/components/schemas/unknownError'
      503:
        $ref: '#/components/responses/x503'
```

7.2.5 Create Encounter

7.2.5.1 Description

The *Create Encounter* service shall, for the specified subject, create a new encounter record and associate an encounter ID to that record. If not provided by the requester, the ***Create Encounter*** service shall generate an encounter ID that uniquely identifies the encounter within the subject record in the system.

Typically, the IAVS service provider will assign the encounter ID. In the event that the requester assigns the encounter ID, it shall be used unless it duplicates an existing encounter ID in which case an error shall be returned.

The *Create Encounter* service is performed prior to a *Set Biographic Data*, *Set Biometric Data*, or *Set Document Data* operation.

NOTE When in encounter mode, for comparison operations it is not necessary to explicitly create an encounter. The IAVS service provider will create the encounter and will set the encounter type to "recognition".

7.2.5.2 REST method

POST

© ISO/IEC 2023 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO/IEC 30108-2:2023

7.2.5.3 Parameters

— *Subject ID* — the identifier of the subject.

— *Encounter Type* — the category of encounter.

— *Encounter ID (optional)* — the identifier of the encounter.

7.2.5.4 Responses

In addition to the generic HTTP responses defined in 9.2, the execution of this service shall provide one of the responses listed in Table 4 (see 9.3 for the meaning of each error condition codes):

Table 4 — Particular responses for *Create Encounter* service

HTTP status code	HTTP status code meaning	Error condition code	Description
200	OK	SUCCESS	Adding also *Encounter ID*, as seen below table.
400	Bad Request	INVALID_SUBJECT_ID	
		INVALID_ENCOUNTER_TYPE	
		INVALID_ENCOUNTER_ID	
404	Not Found	UNKNOWN_SUBJECT	
		UNKNOWN_ENCOUNTER	
500	Internal Server Error	DUPLICATE_ENCOUNTER_ID	
		INTERNAL_DATABASE_ERROR	

A successful execution of this service will provide:

— *Encounter ID* — the identifier of the encounter.

7.2.5.5 YAML specification

To be placed in section #/paths:

```
/createEncounter:
  post:
    summary: Create Encounter
    parameters:
      - name: subjectID
        in: query
        required: true
        schema:
          type: string
      - name: encounterType
        in: query
        required: true
        schema:
          $ref: '#/components/schemas/EncounterCategoryType'
      - name: encounterID
        in: query
        required: false
        schema:
          type: string
    responses:
      201:
        description: Success. Returns encounter ID created
        content:
          application/json:
            schema:
              type: object
```


© ISO/IEC 2023 – All rights reserved
Documents
Document Sharing Hub
Full document details: BS ISO/IEC 30108-2:2023

```
          properties:
            encounterID:
              type: string
              description: The identifier of the encounter
400:
  description: Bad Request
  content:
    application/json:
      schema:
        oneOf:
          - $ref: '#/components/schemas/invalidSubjectId'
          - $ref: '#/components/schemas/invalidEncounterType'
          - $ref: '#/components/schemas/invalidEncounterId'
404:
  description: Not Found
  content:
    application/json:
      schema:
        oneOf:
          - $ref: '#/components/schemas/unknownSubject'
          - $ref: '#/components/schemas/unknownEncounter'
500:
  description: Internal Server Error
  content:
    application/json:
      schema:
        oneOf:
          - $ref: '#/components/schemas/duplicateEncounterId'
          - $ref: '#/components/schemas/internalDatabaseError'
          - $ref: '#/components/schemas/unknownError'
503:
  $ref: '#/components/responses/x503'
```

7.2.6 Create Subject

7.2.6.1 Description

The *Create Subject* service shall create a new subject record and associate a subject ID to that record. The *Create Subject* service shall generate a subject ID that uniquely identifies the subject in the system.

When cross-system uniqueness is required, universally unique identifiers (UUIDs) should be used for Subject IDs.

7.2.6.2 REST method

POST

7.2.6.3 Parameters

None

7.2.6.4 Responses

In addition to the generic HTTP responses defined in 9.2, the execution of this service shall provide one of the responses listed in Table 5 (see 9.3 for the meaning of each error condition code):

© ISO/IEC 2023 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO/IEC 30108-2:2023

Table 5 — Particular responses for *Create Subject* service

HTTP status code	HTTP status code meaning	Error condition code	Description
200	OK	SUCCESS	Adding also *Subject ID*, as seen below table.
500	Internal Server Error	INTERNAL_DATABASE_ERROR	

A successful execution of this service will provide:

— *Subject ID* — the identifier of the subject.

7.2.6.5 YAML specification

To be placed in section #/paths:

```
/createSubject:
  post:
    summary: Create Subject
    responses:
      201:
        description: Success. Returns the subject ID created
        content:
          application/json:
            schema:
              type: object
              properties:
                subjectID:
                  type: string
                  description: The identifier of the subject
      500:
        description: Internal Server Error
        content:
          application/json:
            schema:
              oneOf:
                - $ref: '#/components/schemas/internalDatabaseError'
                - $ref: '#/components/schemas/unknownError'
      503:
        $ref: '#/components/responses/x503'
```

7.2.7 Delete Biographic Data

7.2.7.1 Description

The *Delete Biographic Data* service shall erase all of the biographic data associated with a given subject record. In the encounter-centric model, the service shall erase all of the biographic data associated with a given encounter, and therefore the encounter ID shall be specified. If no encounter ID is specified, or it is null, biographic data will be removed from all encounters. If a gallery is specified, biographic data will be deleted from that gallery only. When deleting data, IAVS implementations may completely erase the information in order to prevent the ability to reconstruct a record in whole or in part, or they may track and record the deleted information for auditing and/or quality control purposes.

7.2.7.2 REST method

DELETE

7.2.7.3 Parameters

— *Subject ID* — the identifier of the subject.

© ISO/IEC 2023 – All rights reserved
Documents
Document Sharing Hub
Full document details: BS ISO/IEC 30108-2:2023

— *Gallery ID (optional)* — the identifier of the gallery or population group from which the biographic information will be deleted.

— *Encounter ID (conditional)* — the identifier of the encounter, required for encounter-centric models.

7.2.7.4 Responses

In addition to the generic HTTP responses defined in 9.2, the execution of this service shall provide one of the responses listed in Table 6 (see 9.3 for the meaning of each error condition code):

Table 6 — Particular responses for *Delete Biographic Data* service

HTTP status code	HTTP status code meaning	Error condition code	Description
200	OK	SUCCESS	
400	Bad Request	INVALID_INPUT	
		INVALID_SUBJECT_ID	
		INVALID_ENCOUNTER_ID	
404	Not Found	UNKNOWN_SUBJECT	
		UNKNOWN_GALLERY	
		UNKNOWN_ENCOUNTER	
500	Internal Server Error	CANNOT_DELETE_DATA	
		INTERNAL_DATABASE_ERROR	

7.2.7.5 YAML specification

To be placed in section #/paths:

```
/deleteBiographicData:
  post:
    summary: Delete Biographic Data
    parameters:
      - name: subjectID
        in: query
        required: true
        schema:
          type: string
      - name: galleryID
        in: query
        required: false
        schema:
          type: string
      - name: encounterID
        in: query
        required: false
        schema:
          type: string
    responses:
      200:
        $ref: '#/components/responses/success'
      400:
        description: Bad Request
        content:
          application/json:
            schema:
              oneOf:
                - $ref: '#/components/schemas/invalidInput'
                - $ref: '#/components/schemas/invalidSubjectId'
                - $ref: '#/components/schemas/invalidEncounterId'
      404:
        description: Not Found
```

© ISO/IEC 2023 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO/IEC 30108-2:2023

```
        content:
          application/json:
            schema:
              oneOf:
                - $ref: '#/components/schemas/unknownSubject'
                - $ref: '#/components/schemas/unknownGallery'
                - $ref: '#/components/schemas/unknownEncounter'
      500:
        description: Internal Server Error
        content:
          application/json:
            schema:
              oneOf:
                - $ref: '#/components/schemas/cannotDeleteData'
                - $ref: '#/components/schemas/internalDatabaseError'
                - $ref: '#/components/schemas/unknownError'
      503:
        $ref: '#/components/responses/x503'
```

7.2.8 Delete Biometric Data

7.2.8.1 Description

The *Delete Biometric Data* service shall remove biometric data from a given subject record. In the encounter-centric model, the encounter ID shall be specified. If no encounter ID is specified, or it is null, biometric data will be removed from all encounters. If a gallery is specified, biometric data will be deleted from that gallery only. If a biometric type(s) is specified, then only biometric data of that type shall be deleted. When deleting data, IAVS implementations may completely erase the information in order to prevent the ability to reconstruct a record in whole or in part, or they may track and record the deleted information for auditing and/or quality control purposes.

7.2.8.2 REST method

DELETE

7.2.8.3 Parameters

— *Subject ID* — the identifier of the subject.

— *Gallery ID (optional)* — the identifier of the gallery or population group from which the biometric information will be deleted.

— *Encounter ID (conditional)* — the identifier of the encounter, required for encounter-centric models.

— *Biometric Type (optional)* — the type of biological or behavioural data to delete, as defined by the BiometricType schema.

7.2.8.4 Responses

In addition to the generic HTTP responses defined in 9.2, the execution of this service shall provide one of the responses listed in Table 7 (see 9.3 for the meaning of each error condition code):

© ISO/IEC 2023 – All rights reserved
Documents
Document Sharing Hub
Full document details: BS ISO/IEC 30108-2:2023

Table 7 — Particular responses for *Delete Biometric Data* service

HTTP status code	HTTP status code meaning	Error condition code	Description
200	OK	SUCCESS	
400	Bad Request	INVALID_SUBJECT_ID	
		INVALID_INPUT	
		INVALID_ENCOUNTER_ID	
		BIOMETRIC_TYPE_NOT_SUPPORTED	
404	Not Found	UNKNOWN_SUBJECT	
		UNKNOWN_GALLERY	
		UNKNOWN_ENCOUNTER	
500	Internal Server Error	CANNOT_DELETE_DATA	
		INTERNAL_DATABASE_ERROR	

7.2.8.5 YAML specification

To be placed in section #/paths:

```
/deleteBiometricData:
  post:
    summary: Delete Biometric Data
    parameters:
      - name: subjectID
        in: query
        required: true
        schema:
          type: string
      - name: galleryID
        in: query
        required: false
        schema:
          type: string
      - name: encounterID
        in: query
        required: false
        schema:
          type: string
      - name: biometricType
        in: query
        required: false
        schema:
          $ref: '#/components/schemas/BiometricType'
    responses:
      200:
        $ref: '#/components/responses/success'
      400:
        description: Bad Request
        content:
          application/json:
            schema:
              oneOf:
                - $ref: '#/components/schemas/invalidSubjectId'
                - $ref: '#/components/schemas/invalidInput'
                - $ref: '#/components/schemas/invalidEncounterId'
                - $ref: '#/components/schemas/biometricTypeNotSupported'
      404:
        description: Not Found
        content:
          application/json:
            schema:
              oneOf:
                - $ref: '#/components/schemas/unknownSubject'
```

© ISO/IEC 2023 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO/IEC 30108-2:2023

```
            - $ref: '#/components/schemas/unknownGallery'
            - $ref: '#/components/schemas/unknownEncounter'
      500:
        description: Internal Server Error
        content:
          application/json:
            schema:
              oneOf:
                - $ref: '#/components/schemas/cannotDeleteData'
                - $ref: '#/components/schemas/internalDatabaseError'
                - $ref: '#/components/schemas/unknownError'
      503:
        $ref: '#/components/responses/x503'
```

7.2.9 Delete Document Data

7.2.9.1 Description

The *Delete Document Data* service shall erase all of the document data of the specified category or categories associated with a given subject record. In the encounter-centric model the service shall erase all of the document data associated with a given encounter, and therefore the encounter ID shall be specified. If no encounter ID is specified, or it is null, document data will be removed from all encounters. If no categories are specified, then all categories (for the specified encounters) shall be deleted. When deleting data, IAVS implementations may completely erase the information in order to prevent the ability to reconstruct a record in whole or in part, or they may track and record the deleted information for auditing and/or quality control purposes.

7.2.9.2 REST method

DELETE

7.2.9.3 Parameters

— *Subject ID* — the identifier of the subject.

— *Encounter ID (conditional)* — the identifier of the encounter, required for encounter-centric models.

— *Document Category (optional)* — the category(ies) of the identity documents to be deleted.

7.2.9.4 Responses

In addition to the generic HTTP responses defined in 9.2, the execution of this service shall provide one of the responses listed in Table 8 (see 9.3 for meaning of each error condition code):

Table 8 — Particular responses for *Delete Document Data* service

HTTP status code	HTTP status code meaning	Error condition code	Description
200	OK	SUCCESS	
400	Bad Request	INVALID_SUBJECT_ID	
		INVALID_ENCOUNTER_ID	
		INVALID_INPUT	
404	Not Found	UNKNOWN_SUBJECT	
		UNKNOWN_ENCOUNTER	
		UNKNOWN_DOCUMENT_CATEGORY	
500	Internal Server Error	CANNOT_DELETE_DATA	
		INTERNAL_DATABASE_ERROR	

© ISO/IEC 2023 – All rights reserved
Documents
Document Sharing Hub
Full document details: BS ISO/IEC 30108-2:2023

7.2.9.5 YAML specification

To be placed in section #/paths:

```
/deleteDocumentData:
  post:
    summary: Delete Document Data
    parameters:
      - name: subjectID
        in: query
        required: true
        schema:
          type: string
      - name: encounterID
        in: query
        required: false
        schema:
          type: string
      - name: documentCategory
        in: query
        required: false
        schema:
          type: string
    responses:
      200:
        $ref: '#/components/responses/success'
      400:
        description: Bad Request
        content:
          application/json:
            schema:
              oneOf:
                - $ref: '#/components/schemas/invalidSubjectId'
                - $ref: '#/components/schemas/invalidEncounterId'
                - $ref: '#/components/schemas/invalidInput'
      404:
        description: Not Found
        content:
          application/json:
            schema:
              oneOf:
                - $ref: '#/components/schemas/unknownSubject'
                - $ref: '#/components/schemas/unknownEncounter'
                - $ref: '#/components/schemas/unknownDocumentCategory'
      500:
        description: Internal Server Error
        content:
          application/json:
            schema:
              oneOf:
                - $ref: '#/components/schemas/cannotDeleteData'
                - $ref: '#/components/schemas/internalDatabaseError'
                - $ref: '#/components/schemas/unknownError'
      503:
        $ref: '#/components/responses/x503'
```

7.2.10 Delete Encounter

7.2.10.1 Description

The *Delete Encounter* service shall delete an existing encounter record from the system. When deleting an encounter, IAVS implementations may completely erase the encounter information in order to prevent the ability to reconstruct a record or records in whole or in part, or they may track and record the deleted information for auditing and/or quality control purposes.

© ISO/IEC 2023 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO/IEC 30108-2:2023

7.2.10.2 REST method

DELETE

7.2.10.3 Parameters

— *Subject ID* — the identifier of the subject.

— *Encounter ID* — the identifier of the encounter.

7.2.10.4 Responses

In addition to the generic HTTP responses defined in 9.2, the execution of this service shall provide one of the responses listed in Table 9 (see 9.3 for the meaning of each error condition code):

Table 9 — Particular responses for *Delete Encounter* service

HTTP status code	HTTP status code meaning	Error condition code	Description
200	OK	SUCCESS	
400	Bad Request	INVALID_SUBJECT_ID	
		INVALID_ENCOUNTER_ID	
404	Not Found	UNKNOWN_SUBJECT	
		UNKNOWN_ENCOUNTER	
500	Internal Server Error	CANNOT_DELETE_DATA	
		INTERNAL_DATABASE_ERROR	

7.2.10.5 YAML specification

To be placed in section #/paths:

```
/deleteEncounter:
  post:
    summary: Delete Encounter
    parameters:
      - name: subjectID
        in: query
        required: true
        schema:
          type: string
      - name: encounterID
        in: query
        required: false
        schema:
          type: string
    responses:
      200:
        $ref: '#/components/responses/success'
      400:
        description: Bad Request
        content:
          application/json:
            schema:
              oneOf:
                - $ref: '#/components/schemas/invalidSubjectId'
                - $ref: '#/components/schemas/invalidEncounterId'
      404:
        description: Not Found
        content:
          application/json:
            schema:
              oneOf:
```


© ISO/IEC 2023 – All rights reserved
Documents
Document Sharing Hub
Full document details: BS ISO/IEC 30108-2:2023

```
                - $ref: '#/components/schemas/unknownSubject'
                - $ref: '#/components/schemas/unknownEncounter'
        500:
          description: Internal Server Error
          content:
            application/json:
              schema:
                oneOf:
                - $ref: '#/components/schemas/cannotDeleteData'
                - $ref: '#/components/schemas/internalDatabaseError'
                - $ref: '#/components/schemas/unknownError'
        503:
          $ref: '#/components/responses/x503'
```

7.2.11 Delete Subject

7.2.11.1 Description

The *Delete Subject* service shall delete an existing subject record and, in an encounter-centric model, any associated encounter information from the system. This service shall also remove the subject from any registered galleries. When deleting a subject, IAVS implementations may completely erase the subject information in order to prevent the ability to reconstruct a record or records in whole or in part, or they may track and record the deleted information for auditing and/or quality control purposes.

7.2.11.2 REST method

DELETE

7.2.11.3 Parameters

— *Subject ID* — the identifier of the subject.

7.2.11.4 Responses

In addition to the generic HTTP responses defined in 9.2, the execution of this service shall provide one of the responses listed in Table 10 (see 9.3 for the meaning of each error condition code):

Table 10 — Particular responses for *Delete Subject* service

HTTP status code	HTTP status code meaning	Error condition code	Description
200	OK	SUCCESS	
400	Bad Request	INVALID_SUBJECT_ID	
404	Not Found	UNKNOWN_SUBJECT	
500	Internal Server Error	CANNOT_DELETE_DATA	
		INTERNAL_DATABASE_ERROR	

7.2.11.5 YAML specification

To be placed in section #/paths:

```
  /deleteSubject:
    post:
      summary: Delete Subject
      parameters:
        - name: subjectID
          in: query
          required: true
          schema:
            type: string
```

© ISO/IEC 2023 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO/IEC 30108-2:2023

```
      responses:
        200:
          $ref: '#/components/responses/success'
        400:
          description: Invalid Subject ID
          content:
            application/json:
              schema:
                $ref: '#/components/schemas/invalidSubjectId'
        404:
          description: Unknown Subject
          content:
            application/json:
              schema:
                $ref: '#/components/schemas/unknownSubject'
        500:
          description: Internal Server Error
          content:
            application/json:
              schema:
                oneOf:
                  - $ref: '#/components/schemas/cannotDeleteData'
                  - $ref: '#/components/schemas/internalDatabaseError'
                  - $ref: '#/components/schemas/unknownError'
        503:
          $ref: '#/components/responses/x503'
```

7.2.12 Delete Subject From Gallery

7.2.12.1 Description

The *Delete Subject From Gallery* service shall remove the registration of a subject from a gallery or population group. The subject shall be identified by either the subject ID or the claim to identity that was specified in the *Add Subject To Gallery* service.

7.2.12.2 REST method

DELETE

7.2.12.3 Parameters

— *Gallery ID* — the identifier of the gallery or population group from which the subject will be deleted.

— *Subject ID (conditional)* — the identifier of the subject; required if an identity claim is not provided.

— *Identity Claim (conditional)* — the identifier by which the subject is known to the gallery; required if a subject ID is not provided.

7.2.12.4 Responses

In addition to the generic HTTP responses defined in 9.2, the execution of this service shall provide one of the responses listed in Table 11 (see 9.3 for the meaning of each error condition code):

© ISO/IEC 2023 – All rights reserved
Documents
Document Sharing Hub
Full document details: BS ISO/IEC 30108-2:2023

Table 11 — Particular responses for *Delete Subject From Gallery* service

HTTP status code	HTTP status code meaning	Error condition code	Description
200	OK	SUCCESS	
400	Bad Request	INVALID_SUBJECT_ID	
		INVALID_INPUT	
		INVALID_IDENTITY_CLAIM	
404	Not Found	UNKNOWN_GALLERY	
		UNKNOWN_SUBJECT	
		UNKNOWN_IDENTITY_CLAIM	
500	Internal Server Error	CANNOT_DELETE_DATA	
		INTERNAL_DATABASE_ERROR	

7.2.12.5 YAML specification

To be placed in section #/paths:

```
/deleteSubjectFromGallery:
  post:
    summary: Delete Subject From Gallery
    parameters:
      - name: galleryID
        in: query
        required: true
        schema:
          type: string
      - name: subjectID
        in: query
        required: false
        schema:
          type: string
      - name: identityClaim
        in: query
        required: false
        schema:
          type: string
    responses:
      200:
        $ref: '#/components/responses/success'
      400:
        description: Bad Request
        content:
          application/json:
            schema:
              oneOf:
                - $ref: '#/components/schemas/invalidSubjectId'
                - $ref: '#/components/schemas/invalidInput'
                - $ref: '#/components/schemas/invalidIdentityClaim'
      404:
        description: Not Found
        content:
          application/json:
            schema:
              oneOf:
                - $ref: '#/components/schemas/unknownGallery'
                - $ref: '#/components/schemas/unknownSubject'
                - $ref: '#/components/schemas/unknownIdentityClaim'
      500:
        description: Internal Server Error
        content:
          application/json:
            schema:
              oneOf:
```

© ISO/IEC 2023 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO/IEC 30108-2:2023

```
                - $ref: '#/components/schemas/cannotDeleteData'
                - $ref: '#/components/schemas/internalDatabaseError'
                - $ref: '#/components/schemas/unknownError'
        503:
          $ref: '#/components/responses/x503'
```

7.2.13 Get Identify Subject Results

7.2.13.1 Description

The *Get Identify Subject Results* service shall retrieve the identification results for the specified token. This service is used in conjunction with the *Identify Subject* service. If the *Identify Subject* service is implemented as an asynchronous service, the implementing system returns a token, and the *Get Identify Subject Results* service is used to poll for the results of the original *Identify Subject* request.

7.2.13.2 REST method

GET

7.2.13.3 Parameters

— *Token* — a value used to retrieve the results of the *Identify Subject* request.

7.2.13.4 Responses

In addition to the generic HTTP responses defined in 9.2, the execution of this service shall provide one of the responses list in Table 12 (see 9.3 for the meaning of each error condition code):

Table 12 — Particular responses for *Get Identify Subject Results* service

HTTP status code	HTTP status code meaning	Error condition code	Description
200	OK	SUCCESS	Adding also *Candidate List*, as seen below table.
400	Bad Request	INVALID_TOKEN	
404	Not Found	TOKEN_EXPIRED	
		IDENTIFICATION_RESULT_NOT_YET_AVAILABLE	
500	Internal Server Error	CANNOT_RETRIEVE_DATA	
		INTERNAL_DATABASE_ERROR	

A successful execution of this service will provide:

— *Candidate List* — a rank-ordered list of candidates that have a likelihood of matching the input biometric sample.

7.2.13.5 YAML specification

To be placed in section #/paths:

```
/getIdentifySubjectResults:
  get:
    summary: Get Identify Subject Results
    parameters:
      - name: token
        in: query
        required: true
        schema:
          $ref: '#/components/schemas/TokenType'
    responses:
```


© ISO/IEC 2023 – All rights reserved
Documents
Document Sharing Hub
Full document details: BS ISO/IEC 30108-2:2023

```
        200:
          description: Success. Returns the ranked Candidate List
          content:
            application/json:
              # a rank-ordered list of candidates that have a
              # likelihood of matching the input biometric sample
              schema:
                $ref: '#/components/schemas/CandidateListType'
        400:
          description: Invalid Token
          content:
            application/json:
              schema:
                $ref: '#/components/schemas/invalidToken'
        404:
          description: Not Found
          content:
            application/json:
              schema:
                oneOf:
                  - $ref: '#/components/schemas/tokenExpired'
                  - $ref: '#/components/schemas/identificationResultNotYetAvailable'
        500:
          description: Internal Server Error
          content:
            application/json:
              schema:
                oneOf:
                  - $ref: '#/components/schemas/cannotRetrieveData'
                  - $ref: '#/components/schemas/internalDatabaseError'
                  - $ref: '#/components/schemas/unknownError'
        503:
          $ref: '#/components/responses/x503'
```

7.2.14 Identify Subject

7.2.14.1 Description

The *Identify Subject* service shall perform an identification search against a given gallery for a given biometric type, returning a rank-ordered candidate list of a given maximum size.

If the *Identify Subject* service is implemented as a synchronous service, the implementing system shall immediately process the request and return the results in the candidate list. If the *Identify Subject* service is implemented as an asynchronous service, the implementing system shall return a token, which is an indication that the request is being handled asynchronously. In this case, the *Get Identify Subject Results* service shall be used to poll for the results of the *Identify Subject* request.

The candidate list shall include only those candidates whose comparison score exceeds a system-defined threshold, thus indicating a likely match with the subject. In the event that there are more candidates than the requested Max List Size, the system shall determine which candidate is included in the last position in the candidate list.

NOTE When in encounter mode, for comparison operations, it is not necessary to explicitly create an encounter. The IAVS service provider will create the encounter and will set the encounter type to "recognition".

7.2.14.2 REST method

GET

7.2.14.3 Parameters

— *BIR* — data structure containing the biometric sample for the search.

NOTE When multiple samples are included as input (e.g. in a multimodal operation), a complex BIR is used.

© ISO/IEC 2023 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO/IEC 30108-2:2023

— *Max List Size* — the maximum size of the candidate list that should be returned.

— *Gallery ID (optional)* — the identifier of the gallery or population group which will be searched; this parameter may also be used to identify an external system where the identification request should be forwarded, if this capability is supported by the implementing system. This parameter shall not be used in conjunction with *Gallery.*

— *Gallery (optional)* — a list of BIRs that shall be used instead of a stored gallery. This parameter shall not be used in conjunction with *Gallery ID.*

7.2.14.4 Responses

In addition to the generic HTTP responses defined in 9.2, the execution of this service shall provide one of the responses listed in Table 13 (see 9.3 for the meaning of each error condition code):

Table 13 — Particular responses for *Identify Subject* service

HTTP status code	HTTP status code meaning	Error condition code	Description
200	OK	SUCCESS	Adding also — *Candidate List* — *Token*, as seen below table.
400	Bad Request	INVALID_BIR	
		INVALID_INPUT	
		BIOMETRIC_TYPE_NOT_SUPPORTED	
		UNKNOWN_FORMAT	
403	Forbidden	BIR_DECRYPTION_FAILURE	
		BIR_QUALITY_ERROR	
		BIR_SIGNATURE_FAILURE	
404	Not Found	UNKNOWN_GALLERY	
		CANNOT_IDENTIFY_DATA	
		IDENTIFICATION_RESULT_NOT_YET_AVAILABLE	
500	Internal Server Error	CANNOT_PROCESS_DATA	
		CANNOT_RETRIEVE_DATA	
		INTERNAL_DATABASE_ERROR	

A successful execution of this service will provide:

— *Candidate List (conditional)* — a rank-ordered list of candidates that have a likelihood of matching the input biometric sample; returned with successful, synchronous request processing.

— *Token* — a token used to retrieve the results of the *Identify Subject* request, returned with asynchronous request processing. If set to zero, operation is processed synchronously and candidate list is returned. If set to a non-zero value, operation is processed asynchronously and *Get Identify Results* needs to be used to retrieve the results.

NOTE Multiple scores/candidates is already built in as a score comes with a CandidateType which is a member of Candidate List.

7.2.14.5 YAML specification

To be placed in section #/paths:

© ISO/IEC 2023 – All rights reserved
Documents
Document Sharing Hub
Full document details: BS ISO/IEC 30108-2:2023

```
/identifySubject:
  get:
    summary: Identify Subject
    parameters:
      - name: bir
        in: query
        required: true
        schema:
          $ref: '#/components/schemas/CBEFF_BIR_Type'
      - name: maxListSize
        in: query
        required: true
        schema:
          type: integer
          format: int32
      - name: galleryID
        in: query
        required: false
        schema:
          type: string
      - name: gallery
        in: query
        required: false
        schema:
          $ref: '#/components/schemas/CBEFF_BIR_ListType'
    responses:
      200:
        description: Success. Returns quality score and algorithm version
        content:
          application/json:
            schema:
              type: object
              properties:
                candidateList:
                  # A rank-ordered list of candidates that have a likelihood
                  # of matching the input biometric sample; returned with
                  # successful, synchronous request processing
                  $ref: '#/components/schemas/CBEFF_BIR_ListType'
                token:
                  # A token used to retrieve the results of the Identify
                  # Subject request; returned with asynchronous request
                  # processing. If set to zero, operation is processed
                  # synchronously and candidate list is returned. If set to a
                  # non-zero value, operation is processed asynchronously and
                  # Get Identify Results must be used to retrieve the results
                  $ref: '#/components/schemas/TokenType'
      400:
        description: Bad Request
        content:
          application/json:
            schema:
              oneOf:
                - $ref: '#/components/schemas/invalidBir'
                - $ref: '#/components/schemas/invalidInput'
                - $ref: '#/components/schemas/biometricTypeNotSupported'
                - $ref: '#/components/schemas/unknownFormat'
      403:
        description: Forbidden
        content:
          application/json:
            schema:
              oneOf:
                - $ref: '#/components/schemas/birDecryptionFailure'
                - $ref: '#/components/schemas/birQualityError'
                - $ref: '#/components/schemas/birSignatureFailure'
      404:
        description: Not Found
        content:
          application/json:
            schema:
              oneOf:
```

© ISO/IEC 2023 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO/IEC 30108-2:2023

```
                - $ref: '#/components/schemas/unknownGallery'
                - $ref: '#/components/schemas/cannotIdentifyData'
                - $ref: '#/components/schemas/identificationResultNotYetAvailable'
        500:
          description: Internal Server Error
          content:
            application/json:
              schema:
                oneOf:
                - $ref: '#/components/schemas/cannotProcessData'
                - $ref: '#/components/schemas/cannotRetrieveData'
                - $ref: '#/components/schemas/internalDatabaseError'
                - $ref: '#/components/schemas/unknownError'
        503:
          $ref: '#/components/responses/x503'
```

7.2.15 List Biographic Data

7.2.15.1 Description

The *List Biographic Data* service shall list the biographic data elements stored for a subject using the Biographic Data Elements output parameter. Note that no actual biographic data is returned by this service (see the *Retrieve Biographic Data* service to obtain the biographic data). In the encounter-centric model, an encounter ID may be specified to indicate that only the biographic data elements stored for that encounter should be returned. If an encounter ID is not specified and encounter data exists for the subject, the service shall return the list of encounter IDs which contain biographic data using the Encounter List output parameter, and the Biographic Data Elements output parameter shall be empty.

7.2.15.2 REST method

GET

7.2.15.3 Parameters

— *Subject ID* — the identifier of the subject.

— *Encounter ID (optional)* — the identifier of the encounter.

— *Encounter Type (optional)* — the category of encounter.

7.2.15.4 Responses

In addition to the generic HTTP responses defined in 9.2, the execution of this service shall provide one of the responses listed in Table 14 (see 9.3 for the meaning of each error condition code):

Table 14 — Particular responses for *List Biographic Data* service

HTTP status code	HTTP ctatus code meaning	Error condition code	Description
200	OK	SUCCESS	Adding also — *Biographic Data Elements* — *Encounter List* as seen below table.
400	Bad Request	INVALID_SUBJECT_ID	
		INVALID_ENCOUNTER_ID	
		INVALID_ENCOUNTER_TYPE	

© ISO/IEC 2023 – All rights reserved
Documents
Document Sharing Hub
Full document details: BS ISO/IEC 30108-2:2023

Table 14 *(continued)*

HTTP status code	HTTP ctatus code meaning	Error condition code	Description
404	Not Found	UNKNOWN_SUBJECT	
		UNKNOWN_ENCOUNTER	
500	Internal Server Error	CANNOT_PROCESS_DATA	
		CANNOT_RETRIEVE_DATA	
		INTERNAL_DATABASE_ERROR	

A successful execution of this service will provide:

— *Biographic Data Elements (conditional)* — a list of biographic data elements associated with a subject or encounter; non-empty if the service was successful, biographic data exists, and either:

a) the person-centric model is being used, or

b) the encounter-centric model is being used and an encounter identifier was specified.

— *Encounter List (conditional)* — a list of encounter IDs associated with a subject and which contain biographic data; non-empty if the service was successful, biographic data exists, the encounter-centric model is being used, and an encounter identifier was not specified

7.2.15.5 YAML specification

To be placed in section #/paths:

```
/listBiographicData:
  get:
    summary: List Biographic Data
    parameters:
      - name: subjectID
        in: query
        required: true
        schema:
          type: string
      - name: encounterID
        in: query
        required: false
        schema:
          type: string
      - name: encounterType
        in: query
        required: false
        schema:
          $ref: '#/components/schemas/EncounterCategoryType'
    responses:
      200:
        description: >
          Success. Returns Biographic Data Elements and/or Encounter List
        content:
          application/json:
            schema:
              type: object
              properties:
                biographicDataElements:
                  # A list of biographic data elements associated with a
                  # subject or encounter; non-empty if the service was
                  # successful, biographic data exists, and either (a) the
                  # person-centric model is being used or (b) the
                  # encounter-centric model is being used and an encounter
                  # identifier was specified
                  $ref: '#/components/schemas/BiographicDataType'
                encounterList:
```

© ISO/IEC 2023 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO/IEC 30108-2:2023

```
                # a list of encounter ID's associated with a subject and
                # which contain biographic data; non-empty if the service
                # was successful, biographic data exists, the
                # encounter-centric model is being used, and an encounter
                # identifier was not specified
                $ref: '#/components/schemas/EncounterListType'
        400:
          description: Bad Request
          content:
            application/json:
              schema:
                oneOf:
                  - $ref: '#/components/schemas/invalidSubjectId'
                  - $ref: '#/components/schemas/invalidEncounterId'
                  - $ref: '#/components/schemas/invalidEncounterType'
        404:
          description: Not Found
          content:
            application/json:
              schema:
                oneOf:
                  - $ref: '#/components/schemas/unknownSubject'
                  - $ref: '#/components/schemas/unknownEncounter'
        500:
          description: Internal Server Error
          content:
            application/json:
              schema:
                oneOf:
                  - $ref: '#/components/schemas/cannotProcessData'
                  - $ref: '#/components/schemas/cannotRetrieveData'
                  - $ref: '#/components/schemas/internalDatabaseError'
                  - $ref: '#/components/schemas/unknownError'
        503:
          $ref: '#/components/responses/x503'
```

7.2.16 List Biometric Data

7.2.16.1 Description

The *List Biometric Data* service shall list the biometric data elements stored for a subject using the Biometric Data Elements output parameter. Note that no actual biometric data is returned by this service (see the *Retrieve Biometric Data* service to obtain the biometric data). In the encounter-centric model, an encounter ID may be specified to indicate that only the biometric data elements stored for that encounter should be returned. If an encounter ID is not specified and encounter data exists for the subject, the service shall return the list of encounter IDs which contain biometric data using the Encounter List output parameter, and the Biometric Data Elements output parameter shall be empty.

An optional parameter may be used to indicate a filter on the list of returned data. Such a filter may indicate that only biometric types should be listed (e.g. face, finger, iris, etc.) or that only biometric subtypes for a particular biometric type should be listed (e.g. all fingerprints: left slap, right index, etc.). If a filter is not specified, all biometric type and biometric subtype information shall be listed (e.g. left index finger, right iris, face frontal, etc.).

7.2.16.2 REST method

GET

7.2.16.3 Parameters

— *Subject ID* — the identifier of the subject.

— *Encounter ID (optional)* — the identifier of the encounter.

— *Encounter Type (optional)* — the category of encounter.

© ISO/IEC 2023 – All rights reserved
Documents
Document Sharing Hub
Full document details: BS ISO/IEC 30108-2:2023

— *List Filter (optional)* — indicates what biometric information should be returned.

7.2.16.4 Responses

In addition to the generic HTTP responses defined in 9.2, the execution of this service shall provide one of the responses listed in Table 15 (see 9.3 for the meaning of each error condition code).

Table 15 — Particular responses for *List Biometric Data* service

HTTP status code	HTTP status code meaning	Error condition code	Description
200	OK	SUCCESS	Adding also — *Biometric Data Elements* — *Encounter List* as seen below table.
400	Bad Request	INVALID_SUBJECT_ID	
		INVALID_ENCOUNTER_ID	
		INVALID_ENCOUNTER_TYPE	
		INVALID_INPUT	
404	Not Found	UNKNOWN_SUBJECT	
		UNKNOWN_ENCOUNTER	
500	Internal Server Error	CANNOT_PROCESS_DATA	
		CANNOT_RETRIEVE_DATA	
		INTERNAL_DATABASE_ERROR	

A successful execution of this service will provide:

— *Biometric Data Elements (conditional)* — a list of biometric data elements associated with a subject or encounter; non-empty if the service was successful, biometric data exists, and either

a) the person-centric model is being used, or

b) the encounter-centric model is being used and an encounter identifier was specified.

— *Encounter List (conditional)* — a list of encounter IDs associated with a subject and which contain biometric data; non-empty if the service was successful, biometric data exists, the encounter-centric model is being used, and an encounter identifier was not specified.

7.2.16.5 YAML specification

To be placed in section #/paths:

```
/listBiometricData:
  get:
    summary: List Biometric Data
    parameters:
      - name: subjectID
        in: query
        required: true
        schema:
          type: string
      - name: encounterID
        in: query
        required: false
        schema:
          type: string
      - name: encounterType
        in: query
```

© ISO/IEC 2023 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO/IEC 30108-2:2023

```
        required: false
        schema:
          $ref: '#/components/schemas/EncounterCategoryType'
      - name: listFilter
        in: query
        required: false
        schema:
          $ref: '#/components/schemas/ListFilterType'
    responses:
      200:
        description: >
          Success. Returns Biographic Data Elements and/or Encounter List
        content:
          application/json:
            schema:
              type: object
              properties:
                biometricDataElements:
                  # A list of biometric data elements associated with a
                  # subject or encounter; non-empty if the service was
                  # successful, biometric data exists, and either (a) the
                  # person-centric model is being used or (b) the
                  # encounter-centric model is being used and an encounter
                  # identifier was specified
                  $ref: '#/components/schemas/BiometricDataListType'
                encounterList:
                  # a list of encounter ID's associated with a subject and
                  # which contain biographic data; non-empty if the service
                  # was successful, biographic data exists, the
                  # encounter-centric model is being used, and an encounter
                  # identifier was not specified
                  $ref: '#/components/schemas/EncounterListType'
      400:
        description: Bad Request
        content:
          application/json:
            schema:
              oneOf:
                - $ref: '#/components/schemas/invalidSubjectId'
                - $ref: '#/components/schemas/invalidEncounterId'
                - $ref: '#/components/schemas/invalidEncounterType'
                - $ref: '#/components/schemas/invalidInput'
      404:
        description: Not Found
        content:
          application/json:
            schema:
              oneOf:
                - $ref: '#/components/schemas/unknownSubject'
                - $ref: '#/components/schemas/unknownEncounter'
      500:
        description: Internal Server Error
        content:
          application/json:
            schema:
              oneOf:
                - $ref: '#/components/schemas/cannotProcessData'
                - $ref: '#/components/schemas/cannotRetrieveData'
                - $ref: '#/components/schemas/internalDatabaseError'
                - $ref: '#/components/schemas/unknownError'
      503:
        $ref: '#/components/responses/x503'
```

7.2.17 List Document Data

7.2.17.1 Description

The *List Document Data* service shall list the document categories stored for a subject using the Document Data Elements output parameter. Note that no other document data is returned by this

© ISO/IEC 2023 – All rights reserved
Documents Document Sharing Hub
Full document details: BS ISO/IEC 30108-2:2023

service (see the *Retrieve Document Data* service to obtain document data by category). In the encounter-centric model, an encounter ID may be specified to indicate that only the document data elements stored for that encounter should be returned. If an encounter ID is not specified and encounter data exists for the subject, the service shall return the list of encounter IDs which contain document data using the Encounter List output parameter, and the Document Data Elements output parameter shall be empty.

7.2.17.2 REST method

GET

7.2.17.3 Parameters

— *Subject ID* — the identifier of the subject.

— *Encounter ID (optional)* — the identifier of the encounter.

— *Encounter Type (optional)* — the category of the encounter.

7.2.17.4 Responses

In addition to the generic HTTP responses defined in 9.2, the execution of this service shall provide one of the responses listed in Table 16 (see 9.3 for the meaning of each error condition code):

Table 16 — Particular responses for *List Document Data* service

HTTP status code	HTTP status code meaning	Error condition code	Description
200	OK	SUCCESS	Adding also — *Document Data Categories* — *Encounter List* as seen below table.
400	Bad Request	INVALID_SUBJECT_ID	
		INVALID_ENCOUNTER_ID	
		INVALID_ENCOUNTER_TYPE	
404	Not Found	UNKNOWN_SUBJECT	
		UNKNOWN_ENCOUNTER	
		NON_EXISTENT_DATA	
500	Internal Server Error	CANNOT_RETRIEVE_DATA	
		INTERNAL_DATABASE_ERROR	

A successful execution of this service will provide:

— *Document Data Categories (conditional)* — a list of document categories associated with a subject or encounter; non-empty if the service was successful, document data exists, and either

a) the person-centric model is being used, or

b) the encounter-centric model is being used and an encounter identifier was specified.

— *Encounter List (conditional)* — a list of encounter IDs associated with a subject and which contain document data; non-empty if the service was successful, document data exists, the encounter-centric model is being used, and an encounter identifier was not specified.

© ISO/IEC 2023 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO/IEC 30108-2:2023

7.2.17.5 YAML specification

To be placed in section #/paths:

```
/listDocumentData:
  get:
    summary: List Document Data
    parameters:
      - name: subjectID
        in: query
        required: true
        schema:
          type: string
      - name: encounterID
        in: query
        required: false
        schema:
          type: string
      - name: encounterType
        in: query
        required: false
        schema:
          $ref: '#/components/schemas/EncounterCategoryType'
    responses:
      200:
        description: >
          Success. Returns Biographic Data Elements and/or Encounter List
        content:
          application/json:
            schema:
              type: object
              properties:
                documentDataCategories:
                  description: >
                    a list of document categories associated with a subject
                    or encounter; non-empty if the service was successful,
                    document data exists, and either (a) the person-centric
                    model is being used or (b) the encounter-centric model
                    is being used and an encounter identifier was specified
                  type: string
                encounterList:
                  # a list of encounter ID's associated with a subject and
                  # which contain document data; non-empty if the service
                  # was successful, biographic data exists, the
                  # encounter-centric model is being used, and an encounter
                  # identifier was not specified
                  $ref: '#/components/schemas/EncounterListType'
      400:
        description: Bad Request
        content:
          application/json:
            schema:
              oneOf:
                - $ref: '#/components/schemas/invalidSubjectId'
                - $ref: '#/components/schemas/invalidEncounterId'
                - $ref: '#/components/schemas/invalidEncounterType'
      404:
        description: Not Found
        content:
          application/json:
            schema:
              oneOf:
                - $ref: '#/components/schemas/unknownSubject'
                - $ref: '#/components/schemas/unknownEncounter'
                - $ref: '#/components/schemas/nonExistentData'
      500:
        description: Internal Server Error
        content:
          application/json:
            schema:
              oneOf:
```


© ISO/IEC 2023 – All rights reserved
Documents
Document Sharing Hub
Full document details: BS ISO/IEC 30108-2:2023

```
                - $ref: '#/components/schemas/cannotRetrieveData'
                - $ref: '#/components/schemas/internalDatabaseError'
                - $ref: '#/components/schemas/unknownError'
        503:
          $ref: '#/components/responses/x503'
```

7.2.18 Perform Fusion

7.2.18.1 Description

The *Perform Fusion* service shall accept either comparison score or comparison decision information and create a fused comparison result. The FusionInformationListType, through the FusionInformationType, provides specific elements for comparison score input and comparison decision input for a single identity, while the FusionIdentityListType provides the ability to submit multiple identities to the Perform Fusion service (see 8.6). The fusion method and processes are left to the implementing system.

7.2.18.2 REST method

GET

7.2.18.3 Parameters

— *Fusion Input* — score or decision input information to the fusion method for each identity.

7.2.18.4 Responses

The *Perform Fusion* service returns one of the return codes defined in Clause 9.

In addition to the generic HTTP responses defined in 9.2, the execution of this service shall provide one of the responses listed in Table 17 (see 9.3 for the meaning of each error condition code):

Table 17 — Particular responses for *Perform Fusion* service

HTTP status code	HTTP status code meaning	Error condition code	Description
200	OK	SUCCESS	Adding also *Match*, as seen below table.
400	Bad Request	INVALID_INPUT	
500	Internal Server Error	CANNOT_PROCESS_DATA	

A successful execution of this service will provide:

— *Match* — indicates the result of the fusion method.

7.2.18.5 YAML specification

To be placed in section #/paths:

```
  /performFusion:
    get:
      summary: Perform Fusion
      parameters:
        - name: fusionInput
          in: query
          required: true
          schema:
            $ref: '#/components/schemas/FusionIdentityListType'
      responses:
        200:
          description: >
```

© ISO/IEC 2023 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO/IEC 30108-2:2023